In [1]:
```
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder,OneHotEncoder
from sklearn.model_selection import train_test_split as tts
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

In [2]:
```
data=pd.read_csv(r"C:\Users\john1\OneDrive\Desktop\Projects worked on\loan_data_1.csv")
data.head(10)
```

Out[2]:

	Unnamed: 0	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	0	LP001003	Male	Yes	1.0	Graduate	No	4583.0	1508.0	128.0	360.0	1.0	Rural	N
1	1	LP001005	Male	Yes	0.0	Graduate	Yes	3000.0	0.0	66.0	360.0	1.0	Urban	Y
2	2	LP001006	Male	Yes	0.0	Not Graduate	No	2583.0	2358.0	120.0	360.0	1.0	Urban	Y
3	3	LP001008	Male	No	0.0	Graduate	No	6000.0	0.0	141.0	360.0	1.0	Urban	Y
4	4	LP001013	Male	Yes	0.0	Not Graduate	No	2333.0	1516.0	95.0	360.0	1.0	Urban	Y
5	5	LP001024	Male	Yes	2.0	Graduate	No	3200.0	700.0	70.0	360.0	1.0	Urban	Y
6	6	LP001027	Male	Yes	2.0	Graduate	NaN	2500.0	1840.0	109.0	360.0	1.0	Urban	Y
7	7	LP001029	Male	No	0.0	Graduate	No	1853.0	2840.0	114.0	360.0	1.0	Rural	N
8	8	LP001030	Male	Yes	2.0	Graduate	No	1299.0	1086.0	17.0	120.0	1.0	Urban	Y
9	9	LP001032	Male	No	0.0	Graduate	No	4950.0	0.0	125.0	360.0	1.0	Urban	Y

In [3]:
```
data.describe()
```

Out[3]:

	Unnamed: 0	Dependents	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	381.000000	373.000000	369.000000	363.000000	373.000000	370.000000	351.000000
mean	190.000000	0.680965	3563.422764	1267.005289	104.914209	340.864865	0.837607
std	110.129469	0.993304	1427.371257	2388.048316	28.484822	68.549257	0.369338
min	0.000000	0.000000	150.000000	0.000000	9.000000	12.000000	0.000000
25%	95.000000	0.000000	2583.000000	0.000000	90.000000	360.000000	1.000000
50%	190.000000	0.000000	3326.000000	830.000000	110.000000	360.000000	1.000000
75%	285.000000	1.000000	4226.000000	2008.000000	127.000000	360.000000	1.000000
max	380.000000	3.000000	9703.000000	33837.000000	150.000000	480.000000	1.000000

In [4]:
```
data.count
```

Out[4]:
```
<bound method DataFrame.count of      Unnamed: 0   Loan_ID  Gender Married  Dependents     Education  \
0             0  LP001003    Male     Yes         1.0      Graduate
1             1  LP001005    Male     Yes         0.0      Graduate
2             2  LP001006    Male     Yes         0.0  Not Graduate
3             3  LP001008    Male      No         0.0      Graduate
4             4  LP001013    Male     Yes         0.0  Not Graduate
..          ...       ...     ...     ...         ...           ...
376         376  LP002953    Male     Yes         3.0      Graduate
377         377  LP002974    Male     Yes         0.0      Graduate
378         378  LP002978  Female      No         0.0      Graduate
379         379  LP002979    Male     Yes         3.0      Graduate
380         380  LP002990  Female      No         0.0           NaN

    Self_Employed  ApplicantIncome  CoapplicantIncome  LoanAmount  \
0              No           4583.0             1508.0       128.0
1             Yes           3000.0                0.0        66.0
2              No           2583.0             2358.0       120.0
3              No           6000.0                0.0       141.0
4              No           2333.0             1516.0        95.0
..            ...              ...                ...         ...
376            No           5703.0                0.0       128.0
377            No           3232.0                NaN       108.0
378            No           2900.0                0.0        71.0
379            No           4106.0                0.0        40.0
380           Yes           4583.0                0.0       133.0

     Loan_Amount_Term  Credit_History Property_Area Loan_Status
0               360.0             1.0         Rural           N
1               360.0             1.0         Urban           Y
2               360.0             1.0         Urban           Y
3               360.0             1.0         Urban           Y
4               360.0             1.0         Urban           Y
..                ...             ...           ...         ...
376             360.0             1.0         Urban           Y
377             360.0             1.0         Rural           Y
378             360.0             1.0         Rural           Y
379             180.0             1.0         Rural           Y
380             360.0             0.0     Semiurban           N

[381 rows x 14 columns]>
```

In [5]:
```
data.isnull()
```

Out[5]:

	Unnamed: 0	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
376	False	False	False	False	False	False	False	False	False	False	False	False	False	False
377	False	False	False	False	False	False	False	False	True	False	False	False	False	False
378	False	False	False	False	False	False	False	False	False	False	False	False	False	False
379	False	False	False	False	False	False	False	False	False	False	False	False	False	False
380	False	False	False	False	False	True	False	False	False	False	False	False	False	False

381 rows × 14 columns

In [6]:
```
data.dropna(inplace=True)
data.drop('Loan_ID',axis=1,inplace=True)
```

In [7]:
```
data.columns
```

Out[7]:
```
Index(['Unnamed: 0', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [8]:
```
data_types = data.dtypes
```

In [9]:
```
data_types
```

Out[9]:
```
Unnamed: 0           int64
Gender              object
Married             object
Dependents         float64
Education           object
Self_Employed       object
ApplicantIncome    float64
CoapplicantIncome  float64
LoanAmount         float64
Loan_Amount_Term   float64
Credit_History     float64
Property_Area       object
Loan_Status         object
dtype: object
```

In [10]:
```
# Convert categorical variables to numerical using LabelEncoder and OneHotEncoder
categorical_cols = ['Gender', 'Married', 'Education', 'Self_Employed', 'Property_Area']
label_encoder = LabelEncoder()
data[categorical_cols] = data[categorical_cols].apply(label_encoder.fit_transform)

# Perform one-hot encoding on the categorical columns
onehot_encoder = OneHotEncoder(drop='first')
encoded_features = onehot_encoder.fit_transform(data[categorical_cols])
loan_data_encoded = pd.concat([data.drop(categorical_cols, axis=1), pd.DataFrame(encoded_features.toarray())], axis=1)
```

In [11]:
```
X=data.drop('Loan_Status',axis=1)
y=data['Loan_Status']
```

In [12]:
```
X_train,X_test,y_train,y_test=tts(X,y,test_size=0.3,random_state=42)
```

In [16]:
```
model = LogisticRegression()
model.fit(X_train, y_train)
```

```
C:\Users\john1\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:460: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[16]:
```
LogisticRegression()
```

In [17]:
```
y_pred=model.predict(X_test)
```

In [18]:
```
accuracy=accuracy_score(y_test,y_pred)
accuracy
```

Out[18]:
```
0.8170731707317073
```

In [ ]:
```
```